AMENDMENTS TO THE SPECIFICATION

Please amend the specification, as follows:

Replace paragraph [0018] with the following amended paragraph [0018]:

At time t, a filter cell coefficient vector C(t), which is known as a tap coefficient or a coefficient for the equalizer, may be expressed by ~~expression~~ <u>Expression</u> (2):

$$C(t) = C(t-1) + K(t) \cdot e(t). \qquad \ldots (2)$$

In Expression (2), e(t) denotes a difference between a signal output from the channel equalizer and a training signal that is known to a receiving side (or the channel equalizer), at time t, and K(t) denotes a Kalman gain that is expressed by Expression (3):

$$K(t) = [\lambda^{-1} \cdot P(t-1) \cdot D(t)]/[1 + \lambda^{-1} \cdot D(t)^T \cdot P(t-1) \cdot D(t)]. \qquad \ldots (3)$$

In Expression (3), $0.9 < \lambda < 1$, and P(t-1) denotes an error covariance matrix. The error covariance matrix P(t) is expressed by Expression (4):

$$P(t) = \lambda \cdot P(t-1) - \lambda \cdot K(t) \cdot D(t)^T \cdot \underline{P}(t-1). \qquad \ldots (4)$$

Replace paragraph [0019] with the following amended paragraph [0019]:

A coefficient for a filter cell (or tap) may be updated using the Kalman algorithm, ~~expression~~ as shown by the following equations of Expression (5):

$$\cancel{C(t) = N + (1xN) \cdot C}$$

$$\cancel{K(t) = [C \cdot (NxN) \cdot (Nx1)]/[(I+C) \cdot (1xN) \cdot (NxN) \cdot (Nx1)]}$$

$$\cancel{P(t) = C \cdot (NxN) - C\ (Nx1) \cdot (1xN) \cdot (NxN). \quad \ldots (5)}$$

$$\underline{C(t) = N + (1xN) \cdot C}$$

$$\underline{K(t) = [C \cdot (NxN) \cdot (Nx1)]/[(1+C) \cdot (1xN) \cdot (NxN) \cdot (Nx1)]}$$

$$\underline{P(t) = C \cdot (NxN) - C \cdot (Nx1) \cdot (1xN) \cdot (NxN). \quad \ldots (5)}$$

In Expression (5), C denotes a constant, N denotes the sum of the number of feedforward filter cells and the number of feedback filter cells, and (N×N) denotes a matrix consisting of N ~~matrices~~ columns of the sum and N rows of the sum.

Replace paragraph [0030] with the following amended paragraph [0030]:

When a data sequence DS_i is transmitted to the filtering circuit 200, the filtering circuit 200 receives values $DS_i(t)$ of a data sequence DS_i at a time t. That is, the data sequence DS_i is transmitted to the filtering circuit 200, and the data sequence values $DS_i(t)$ are stored in the filter cells 210_1 through <u>210_m</u> [[210m]] and the filter cells <u>220_1</u> [[2201]] through <u>220_n</u> [[220n]]. At time t, the data sequence value $DS_i(t)$ is transmitted to filtering circuit 200 and stored in data register 230_1 of the first filter cell 210_1.

Replace paragraph [0032] with the following amended paragraph [0032]:

At time t+2, the data sequence value $DS_i(t)$ is transmitted to the data register (not shown) of the third filter cell 210_3 from the data register of the second filter cell 210_2, and the data sequence value $DS_i(t+1)$ is transmitted to the data register of the second filter cell 210_2 from the data register <u>230_1</u> [[2301]] of the first filter cell <u>210_1</u> [[210_2]]. At the same time, a next data sequence value $DS_i(t+2)$ is transmitted to the filtering circuit 200 and stored in the data register 230_1 of the first filter cell 210_1.

Replace paragraph [0033] with the following amended paragraph [0033]:

At time t, each of the multipliers 250_1 through 250_m, of the respective filter cells 210_1 through 210_m, receives a value of a coefficient CK stored in the corresponding coefficient register 240_1 through 240_m, and receives a data value stored in the corresponding data register 230_1 through $\underline{230_m}$ [[230_n]]. Also, each of the multipliers 280_1 through 280_n, of the respective filter cells 220_1 through 220_n, receives a value of a coefficient CK stored in the corresponding coefficient register 270_1 through 270_n, and receives a data value stored in the corresponding data register 260_1 through 260_n. Each of the multipliers 250_1 through 250_m and the multipliers 280_1 through 280_n multiplies the received two values and provides the multiplication result to the adder 290.

Replace paragraph [0037] with the following amended paragraph [0037]:

The DSP 370 receives signals output from comparator 360, Kalman gain memory 320, error covariance memory 330, data memory 340, training sequence memory 350, and adder 290. The DSP 370 processes a received training sequence TS_i, calculates the optimum value of a particular one of coefficients CK for filter cells, and outputs the calculation result to a decision feedback equalizer (DFE) input data memory 380. The DFE input data memory 380 receives signal $(X_{out})'$ from the DSP 370 and outputs $\underline{\text{output}}$

signal X_{out} to the data registers 260_1 through 260_n of the filter cells 220_1 through 220_n.

Replace paragraph [0048] with the following amended paragraph [0048]:

The error calculator 393 calculates a difference, e(t), between the output of the multiplexer 397 and the output of adder 290, and transmits the difference e(t) to a ~~multiplexer~~ multiplier 391. The slicer 395 changes the output of the adder 290 into a value approximately equal to the original transmission signal. The DFE input data memory 380 receives the output of the multiplexer 397 and provides it to data registers 260_1 through 260_n of corresponding filter cells 220_1 through 220_n, for example.

Replace paragraph [0049] with the following amended paragraph [0049]:

A ~~multiplexer~~ multiplier 391 multiplies the difference e(t) output from the error calculator 393 and an output of the Kalman gain calculator 325 and sends the multiplication result to an adder 389. The adder 389 adds the multiplication result output from the ~~multiplexer~~ multiplier 391 to a new coefficient vector stored in the coefficient allocation memory 387, and outputs the addition result to the coefficient memory 310. In this case, the coefficient memory 310 receives a signal output from the adder 389, updates the

coefficient vector using the signal, and stores the updated coefficient vector. In accordance with the exemplary embodiments, modified Kalman algorithms may be used to minimize the values of error signals when updating coefficients.

Replace paragraph [0059] with the following amended paragraph [0059]:

In this case, data signal D(t) = ~~[4×11]~~ [4x1] can be described by ~~expression~~ Expression (14):

$$D(t) = [D1, D2, D7, D10]^T. \quad \ldots (14)$$

Replace paragraph [0060] with the following amended paragraph [0060]:

During a training period, the delay time of a multi-path can be estimated with reference to a value stored in the tap allocation memory 381 [[383]], and it is possible to continuously reduce the number of filter cells to be updated by operating the comparator 360 again, after a lapse of time corresponding to the delay time of the multipath. In this way, the amount of calculation spent on updating coefficients may be reduced, thereby potentially reducing total power consumption.